DOCUMENT RESUME

ED 366 306 IR 016 502

AUTHOR	Vitale, Michael R.; Romance, Nancy
TITLE	Developing Applications of Artificial Intelligence Technology To Provide Consultative Support in the Use of Research Methodology by Practitioners.
PUB DATE	Apr 93
NOTE	20p.; Paper presented at the Annual Meeting of the American Educational Research Association (Atlanta, GA, April 12-16, 1993).
PUB TYPE	Reports - Evaluative/Feasibility (142) -- Speeches/Conference Papers (150)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	Access to Information; *Artificial Intelligence; *Computer Software Development; *Consultation Programs; Educational Technology; *Expert Systems; Knowledge Level; *Researchers; *Research Methodology; Research Utilization

ABSTRACT

Adopting perspectives based on applications of artificial intelligence proven in industry, this paper discusses methodological strategies and issues that underlie the development of such software environments. The general concept of an expert system is discussed in the context of its relevance to the problem of increasing the accessibility of expert assistance to research practitioners. Important methodological issues and development strategies that underlie the construction of such systems are discussed. Some illustrative examples addressing the question of representation of research expertise (versus textbook-based information) are presented. Finally, implications and priorities for the longitudinal development of practitioner-supporting expert systems for the representation of the research knowledge-base and the design of knowledge-based instructional environments for improving graduate training in research are noted. Nine figures illustrate the discussion. (Contains 25 references.) (SLD)

ED 366 306

DEVELOPING APPLICATIONS OF ARTIFICIAL INTELLIGENCE TECHNOLOGY TO PROVIDE CONSULTIVE SUPPORT IN THE USE OF RESEARCH METHODOLOGY BY PRACTITIONERS [1]

Michael R. Vitale, East Carolina University
Nancy Romance, Florida Atlantic University

Abstract

Artificial intelligence methodology offers a potential means for insuring "real time" accessibility of consultive research expertise to research practitioners. Adopting perspectives based upon applications of artificial intelligence applications proven in industry, this paper discusses methodological strategies and issues that underlie the development of such software environments.

Expert systems (Payne & McArthur, 1990) and related techniques (Carr, 1992) are applications of artificial intelligence that have been developed specifically to facilitate the transfer and accessibility of knowledge (i.e., expertise) from human experts to practitioners. The purpose of this paper is to explore methodological issues pertaining to the application of expert systems to the problem of providing consultive research expertise to research practitioners. In doing so, the paper consists of four sections. First, the general concept of an expert system is overviewed in the context of its relevance to the problem of increasing the accessibility of expert assistance to research practitioners. Second, important methodological issues and development strategies that underlie the construction of such systems are discussed. Third, some illustrative examples addressing the question of representation of research expertise (vs. textbook-based information) are presented. And, fourth, the implications (and priorities) for the longitudinal development of practitioner-supporting expert systems, for the representation of the research knowledge-base in the field, and for the design of knowledge-based instructional environments for improving graduate training in research are noted.

Expert Systems: Concepts and Applications for Increasing the Accessibility of Research Expertise

Expert systems are specialized software applications with which practitioners needing advice and guidance on specific problems can interact. The basic requirement for a well-designed expert system in such settings is that it successfully emulates the consultive assistance of a human expert. This perspective would suggest that the ideal circumstances might be those in which human experts with specialized proficiency in research methodology are always available on demand to provide guidance to non-technically oriented research practitioners. The need for expert systems then is implied by the fact that there are far too few research-methods experts to provide the ongoing level of support needed for technically-sound and efficiently-designed research in school settings by inexperienced practitioners, particularly in light of present trends toward action and school-based research.

The emphasis in the present paper is primarily upon the potential of expert systems as tools that provide consultive assistance. However, the development of any expert system requires standards of practice within a discipline that provide the foundation for the forms of procedural knowledge that are

[1] *Paper presented at the Annual Meeting of the American Educational Research Association. Atlanta GA, April, 1993.*

1701682

incorporated into such systems. Thus, in addition to being a tool, expert systems also provide a context for motivating a restructuring of the base of knowledge within a discipline, particularly with regard to integrating textbook-based knowledge with the advanced practices of experts. With this in mind, this paper also notes implications of expert systems methodology for advancing paradigmatic perspectives (e.g., Kuhn, 1970) of knowledge within the field of research itself.

Elaboration of the Concept of an Expert System Consultant. The utilization of an expert system (see Figure 1) by a research practitioner is equivalent to an interactive consultation with a highly experienced human expert to gain assistance in addressing a research problem (e.g., What should I do to conduct research on this topic?, What should my research questions be?) or in solving a specific problem within a research study (e.g., Given the conclusions I want to make, how should I design my study?, Given my study, how should I measure the outcomes?). During such a consultation, the research expert and practitioner would interact to describe the problem, clarify the specific details and constraints, and then specify the actions to be taken. In guiding the interaction, the expert's knowledge of research principles in conjunction with practical rules of thumb based upon extensive research experience would be linked to the practitioners specific knowledge of the problem solution. As part of this process, the consultant would be able to answer questions explaining why certain information is needed, how various conclusions were reached, and how possible changes in the research setting (e.g., research questions, practical constraints) might require alternate solutions (i.e., what if questions).

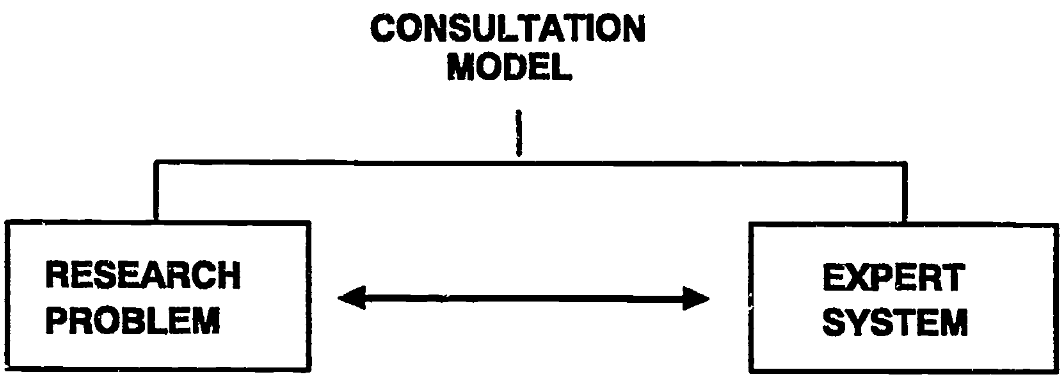
Within an expert systems application, research practitioners would obtain similar consultive assistance through expert systems software developed to emulate the competence of such an experienced human expert (Vitale, 1992). Under conditions in which the availability of human experts is limited, the potential value of expert systems software is the capability to deliver consultive expertise to an individual with a degree of frequency or level of depth that a human expert could not provide due to limited accessibility. In developing such software (see Figure 2), the major logical requirements that must be addressed technically are the representation of research expertise (i.e., the "knowledge-base"), the specification of knowledge-processing mechanisms that control the linking of the knowledge to the specifics of the problem setting in order to identify appropriate courses of action (i.e., the "inference engine"), and the design of the interface through which the researcher practitioner and software will interact.

General Characteristics of Expert System Software Environments. An important step in the development of expert systems software is the selection of a software environment for building (and delivering) the system. Although any general purpose programming language, in principle, could be used, specially designed software development environments called "shells" are commonly used because they facilitate rapid prototyping and efficient system construction for a variety of methodological reasons. Among the most important is that the architecture of the shells separates the "knowledge" component of the software from its other parts (e.g., user interface, management of consultive data, explanation facility, knowledge-processing/inference component). By using shells, developers of expert systems applications are free to focus upon the encoding of expertise (as knowledge) that allows the software to emulate the performance of an expert rather than writing computer program(s) to implement all the various components of the system. Thus, the shell itself provides the developer with a supportive environment containing all of the tools needed to produce the system with the exception of the knowledge (that the developer inserts into the system) and the nature of the interface (about which the developer decides).

Some Technical Aspects of Expert Systems: Knowledge Representation and Logical Inference. When encoding expertise into the shell software, the simplest form of knowledge representation consists of production rules (see Figure 3) in the form of IF/THEN statements through which actions (i.e., THENs) are tied to patterns of conditional information (i.e., IFs). With such knowledge-based rules entered into the software shell, the expert system is able to decide what actions to recommend for problem-solving during a consultation by querying the user regarding circumstances specific to a particular setting. By matching patterns of circumstances to patterns of production rule conditions (IFs) and by following logical

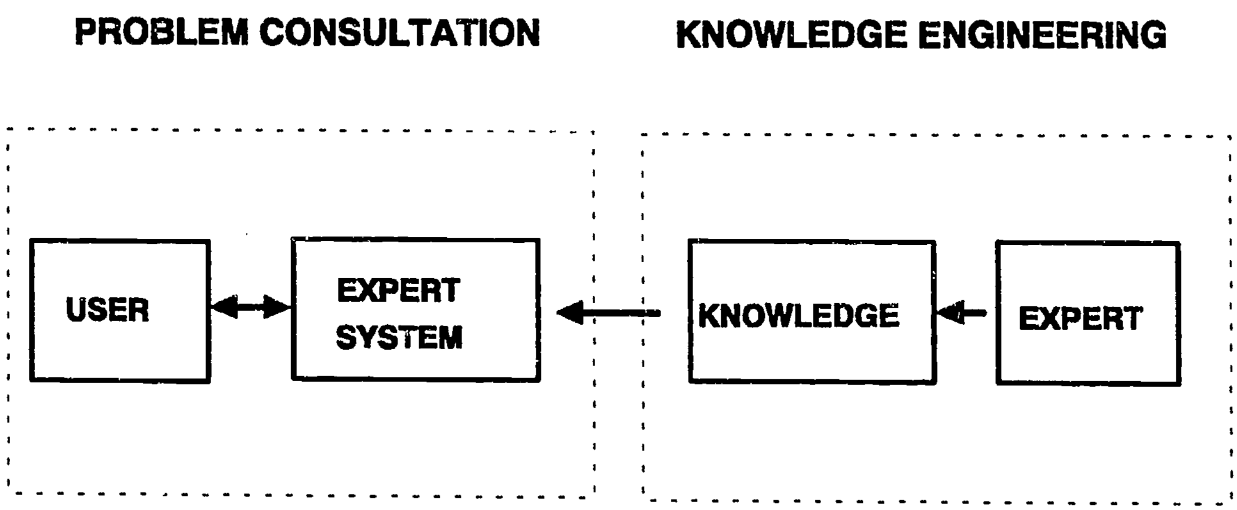
VITALE
AERA 1993
Figure 1

EXPERT SYSTEMS AS CONSULTANTS

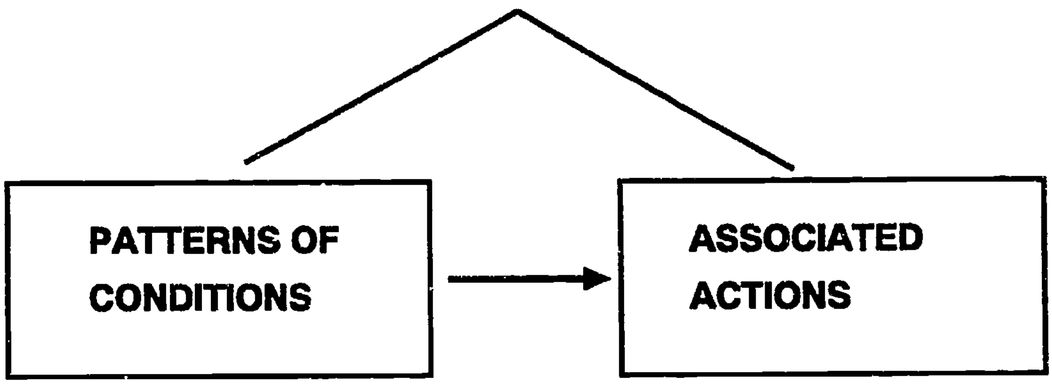


VITALE
AERA 1993
Figure 2

DEVELOPING EXPERT SYSTEMS AS CONSULTANTS



KNOWLEDGE AS RULES



[1] IF < A,B,C > THEN < X,Y >

[2] IF < B,D > THEN < Z >

[3] IF < E > THEN < Z,W >

chains of such rules, the expert systems software can focus upon obtaining relevant information (and ignoring irrelevant details) while determining a proper course of action. In turn, the actions recommended by the program provide guidance for the end-user.

Although the overview in the above paragraph provides a sound general overview, there are several aspects of expert systems architecture that should be noted. First, the logical control mechanisms associated with expert systems (i.e., the inference engine) are an area of continuing research and development. However, within limitations of most obtainable (i.e., usable but affordable) expert systems shells, the major inference mechanisms consist of forward chaining or backward chaining (or combinations of both) applied to an IF/THEN rule base in which the THEN component of the rule is considered alternately as a conclusion, goal, or recommended action (see Figure 4). In forward chaining, the system ordinarily takes a set of situation-specific information as input and then selectively matches (i.e., "fires") IF/THEN rules that generate new conclusions for the system to use. In turn, the rule matching-firing process continues iteratively until the system is successful (or unsuccessful) in producing a satisfactory conclusion or goal in the form of a recommended action.

By way of contrast, in backward chaining the system selects a possible conclusion, goal, or action and then tries to determine if the pattern of conditions (IFs) associated with the conclusion (THEN) is true. In doing so, either the system attempts to prove the conditions by treating them as intermediate goals using the rule-base within the system or by querying an individual for specific input. Failing to confirm a tentative goal, a backward chaining system simply keeps selecting new goal alternatives until a conclusion is proved or until the possible goals (or possible actions) are exhausted. Typically, backward chaining systems are appropriate for diagnostic or classification problems while forward chaining systems are more commonly used for planning or process applications. Alternately, hybrid systems utilize both approaches depending upon the specific circumstances associated with the logical structure of solving a problem.

A second important methodological concern with expert systems has to do with the question of how the knowledge (or expertise) is represented structurally. The technically simplest form of knowledge representation, IF/THEN production rules as described above, are supported by virtually all expert systems shells. And, although production rules can be used to represent virtually any form of expertise, they can become increasingly awkward to develop (and maintain) for more complex forms of expertise. One methodological solution to this problem has been to develop forms of knowledge representation schemes consisting of patterned information. For example, "frames" are complex data structures that consist of "slots" within which information can be entered. An advantage of frames is that they define the relationships among the specific information items entered into slots (which may other frames as well as information items) thereby reducing the overhead for the system developer by eliminating the necessity for representing such relational information in separate production rules.

A second (and related) style of knowledge representation in expert systems focuses on complex (frame-type) structures called "cases." The advantage of case representations (e.g., Riesbeck & Schank, 1989) in expert systems development is that they allow the direct encoding of information in terms of meaningful problem templates in the form of typical case examples. In turn, such case structures are combined with "case-based reasoning" control mechanisms that match patterns of information describing a specific problem with available cases in order to determine patterns of actions. (Near matches result in enhancements to basic cases or the addition of new cases to the knowledge base.) The advantage of case-based reasoning approaches is that they are ordinarily far more efficient to construct and maintain. A practical disadvantage, however, is that case-based models (unlike frames) are not supported by affordable expert systems shells at the present time. Yet, despite this, in using existing shells, it is very much worth the effort to apply the organizational perspectives suggested by these approaches when encoding expertise in the form of production rules. More complete discussions of expert systems methodology and related topics can be found in Luger & Stubblefield (1989) and Winston (1992).

VITALE
AERA 1993
Figure 4

EXPERT SYSTEM LOGIC

FORWARD CHAINING

MATCH CONDITIONS TO ACTIONS
UNTIL GOAL REACHED

BACKWARD CHAINING

PROVE CONDITIONS GIVEN
GOAL

RULES: [1] A&C-->D [2] B--->C [3] A--->B

FORWARD CHAINING	WORKING MEMORY
PROOF:	GIVEN: GOAL IS D A IS TRUE
[1] A--> B	SO: B IS TRUE
[2] B--> C	SO: C IS TRUE
[3] A & C--> D	SO: D IS TRUE

BACKWARD CHAINING	WORKING MEMORY
PROOF:	GIVEN: GOAL IS D
[1] A & C--> D	SO: SUBGOAL IS A SUBGOAL IS C
[2] SINCE NO INFORMATION ON A, ASK: T/F (ASSUME T)	SO: A IS TRUE
[3] SINCE B-->C	SUBGOAL IS B
[4] SINCE NO INFORMATION ON B, ASK: T/F (ASSUME T)	SO: B IS TRUE SO: C IS TRUE SO: D IS TRUE

SIMPLE EXPERT SYSTEM SHELL

.... VP EXPERT VER. 2.0 ....

```
ENDOFF;
ACTIONS
DISPLAY "PRESS RETURN TO BEGIN~"
FIND FRUIT
DISPLAY "FRUIT IS- {FRUIT}";
DISPLAY "SO... PLACE IT IN THE BOX MARKED {FRUIT}";

!RULES

RULE 1
IF      SKIN = SOFT   AND
        TEXTURE = CRUNCHY
THEN  FRUIT = APPLE;
RULE 2
IF      SKIN = SOFT   AND
        TEXTURE = MUSHY
THEN  FRUIT = BANANNA;
RULE 3
IF      SKIN = HARD   AND
        TEXTURE = CRUNCHY
THEN  FRUIT =COCONUT;
RULE 4
IF      SKIN = HARD   AND
        TEXTURE = MUSHY
THEN  FRUIT =UNKNOWN;
```

VP EXPERT COMPONENTS

KNOWLEDGE BASE
 	
PROGRAM CONTROL
.... ACTIONS
.....INFERENCE ENGINE
 	
USER INTERFACE
.... SCREEN DISPLAYS
.... QUEST./RESPONSES
.... OTHER
    .... WHY
    .... HOW
    .... WHAT IF
    .... TRACE/TREE

```
!ASK, CHOICE SECTION

ASK SKIN: "WHAT IS THE SKIN LIKE?";
CHOICES SKIN: SOFT, HARD;

ASK TEXTURE: "WHAT IS THE TEXTURE INSIDE?";
CHOICES TEXTURE: CRUNCHY, MUSHY;
```

Expert Systems Development Strategies and Representative Software Environments. The development of an expert system follows a general strategy for capturing human expertise that is commonly referred to as "knowledge engineering". First, one or more experts are carefully studied across a representative range of problem-solving applications. The purpose here is to determine what the expert does in terms of actions, to describe the conditions under which these different actions are done, and to obtain clarifying explanations from the expert as to why different actions are done under various conditions. Second, the resulting expertise is represented in IF/THEN production rules (or a more advanced knowledge representation scheme) and input into the expert system shell (using a visually-oriented or text-oriented editor). Third, the system is applied to a variety of representative problem situations, the results compared with those of experts, and the knowledge-base (e.g., production rules) modified as necessary. Fourth, the revision-test cycle is repeated until the software can emulate the consultive performance of the expert. And, last, the software is delivered for use after which it's performance in the field is monitored on an ongoing basis so that it can be regularly maintained (e.g., updated, improved). Although additional detail is beyond the scope of the present paper, Payne & McArthur (1990) describe the application of development strategies using a variety of expert system shells and a monthly trade magazine, AI Expert, regularly reviews expert systems software (e.g., Shaw & Zeichick, 1992) for popular microcomputer platforms (IBM, Macintosh).

Methodological Issues in the Development of Expert Systems that Support the Practice of Research

This section overviews key issues specifically relating to the development of expert systems software that would provide consultive assistance to research practitioners.

Developing the Knowledge-Base that Represents Research Expertise. Although textbooks on research methods provide an initial point for determining research expertise, they do not provide an adequate foundation for developing a research knowledge-base for the "knowledge-engineer." This is true for a number of reasons. First, recognizing Anderson's (1982) distinction, expertise must be represented as procedural knowledge (i.e., as actions that occur under patterns of conditions) in order to be usable in an expert system. In this regard, much of the knowledge in textbooks is in declarative form (i.e., knowledge about...) that, even with appropriate translations, is not detailed enough to represent the range of expertise needed by the practicing researcher. Second, the scope of expertise needed to conduct research (e.g., Jackson, 1990) is far broader than that encompassed by traditional textbooks in research methods. In effect, despite their value in representing the knowledge-base in some areas (e.g., statistical analysis of data, aspects of experimental design, aspects of measurement/data collection procedures), textbooks as a whole are incomplete representations of the broad range of expertise required to conduct sound research.

Because textbook-based knowledge is of limited value, the full development of a research knowledge-base can only result from the analysis of the performance of experts working on a representative range of research problems. With this need in mind, Vitale (1993) has presented a scheme for classifying empirical research studies that could provide the basis for specifying the necessary problem-domain in terms of the stages through which scientific knowledge naturally evolves. The resulting continuum then could provide sufficient structure in the form of a domain to facilitate the explication of categories of research expertise that focus on the following:

* establishing the existence of a phenomenon (as a classificatory category)
* determining the conditions under which a phenomenon occurs (for reliable observation of its occurrence)
* relating the phenomenon to other phenomena (for predictive understanding)

* determining manipulatable conditions that affect the occurrence of the phenomenon (for understanding in terms of scientific control)

* developing theory in the form of symbolic codification of scientific knowledge (to allow representation and communication of the above within the scientific community)

* engineering technological applications that incorporate theory (devising ways of applying theory considered as scientific knowledge)

* verifying the soundness of technological applications (to determine whether an application is useful and to evaluate associated implications for its theoretical foundations)

Complementing this scheme, Vitale (1993) argued for mapping the interrelationship of various types of research perspectives (e.g., qualitative, quantitative, applied, theoretical, action, etc.) in terms of a two underlying dimensions: (a) an absolute distinction between experimental and non-experimental research in combination with (b) a continuum of research settings ranging from artificial (or laboratory) to naturalistic.

From the standpoint of organizing research problems (and expertise), the idea that either experimental or non-experimental research (considered to allow qualitatively different forms of conclusions) may be conducted in either naturalistic or laboratory settings and the idea that the domain of all research (including the scheme above) is encompassed by these two classification dimensions has much to recommend it with regard to parsimony. In this spirit (after Sidman, 1960), this scheme considers all research as theoretical, with some theoretical research also being applied _if_ it addresses behaviors having societal value; while (after Johnson & Pennypacker, 1980), the concerns of qualitative researchers are repositioned to fall within a broadened perspective having to do with measurement and scaling (within which traditional data analysis practices also fall).

In particular, from the standpoint of addressing the question of explicating the research expertise required by the research practitioner, Vitale's (1993) scheme implies that qualitative (e.g., Bogdan & Biklin, 1992; Marshall & Rossman, 1989; Tesch, 1990) and quantitative research (e.g., Kerlinger, 1986) may be represented as part of a broader research perspective (e.g., Johnson & Pennypacker, 1980; Sidman, 1960), and, thus, do not represent fundamentally different distinctions because they do not require different domains of expertise. Rather, this distinction is considered to denote personal characteristics of researchers themselves that need not be encompassed within expertise; in combination with objectively-defined methodological preferences, practices, and styles that are able to be "captured" through knowledge-engineering and incorporated into a common domain of research expertise. In any case, whether Vitale's (1993) scheme or an alternate scheme is used to represent the domain of research practice, the performance of expert researchers across the domain must be represented fully in the knowledge-base of the expert system in order for it to provide effective guidance through consultation. In this regard, as textbooks on research methods are clearly inadequate in both breath and depth, some form of knowledge engineering strategies are necessarily required.

Flexible Development Strategies and Software Environments. Although the process for developing an expert system in general and the knowledge-base in particular has been presented as if it consists of a sequential series of discrete steps, in most cases it does not. Rather, the development of an expert system is much more of a "bootstrap" operation in which different parts of the development process interact with each another. This is because the knowledge of most experts comprise what are known as "ill-defined domains." As a result of what might be considered "true" expertise being very difficult to capture directly, the development of most expert systems emphasizes incremental prototyping (vs top-down design) as a development strategy in which increasingly larger parts of the system are implemented and evaluated until the final system is completed.

In following a prototyping strategy, the selection of a software development environment (e.g., shell) is critical since extensive editing and restructuring of the knowledge-base may be required at any point in the development process and since the size of the system cannot be determined in advance. Therefore, before initiating a project using a shell, it is important to evaluate the software environment in terms of these concerns and, if possible, talk directly to individuals (i.e., knowledge-engineering experts) who have prior experience developing the type of application that is being considered regarding these and other technical concerns (e.g., type of inference engine(s) supported, object-oriented knowledge structures). Although beyond the scope of this paper, some good expert systems shells to investigate for IBM (and compatible) microcomputer platforms (486-33 architecture or above with 16MB RAM and large/fast hard drives (2?0MB/12MS) highly recommended) running under Microsoft Windows 3.0 or above are Symbologic Adept, Goldworks III, Level 5 Object, and EXSYSS.

In many cases, the application domain, the associated expertise, and the needs of the end-user are so ill-defined from the standpoint of being implemented through software that substantial preliminary development work must be done in order to clarify the consultive problem being addressed. Our feeling is that the question of providing consultive assistance to practitioners through expert systems is clearly in this category. As a result, our present efforts in developing consultive research systems are pursuing the following strategy:

* an initial attempt to describe the domain of research and research expertise in a parsimonious and modular form
* an explication of the consultive needs of practitioners having little research experience
* the use of a hypertext-type of software environment (e.g., Toolbook on the IBM platform) that allows some aspects of the potential knowledge base and the means of communication of information between software and user to be explored informally
* the initial prototyping of modular elements of the expert system using Symbologic Adept which provides a unique visual programming environment in which procedures as sequences of activities can be represented directly
* the subsequent development of the system using a knowledge-based expert system (e.g., Goldworks III) that supports production rules as part of more advanced forms of integrated knowledge structures (e.g., frames, objects) and alternate forms of inference (e.g., backward chaining, forward chaining, hybrid models).

Overall, the adoption of the above strategy incorporating the use of a number of complementary software environments was based upon a number of considerations. First, a major underlying concern was the assumption that the explication of the research knowledge-base in conjunction with the needs of the research practitioner would likely involve substantial revision. This is simply due to the fact that not all aspects of educational research methodology are well developed (see following section) and that the many areas that are well-developed do not always represent methodological practices in ways that are translatable into expertise in the form of capabilities or procedures. Second, some interactive field-testing is required to develop a workable interface model. This is far easier to do in an informal prototyping fashion using a hypertext environment such as Toolbook, before encoding complex forms of knowledge into an expert system shell.

The third consideration is important because we anticipate that an emphasis on modularity and the capability to integrate different software environments may facilitate the overall development effort and provide a more naturally-usable form of consultive software for research practitioners. For example, during a consultation it might be natural for the user to browse and select different alternatives- a capability that is much easier to develop in a hypertext environment such as Toolbook. Or, some procedures might be developed most easily in Symbologic Adept because it is a form of shell that is designed to allow procedural knowledge to be naturally encoded. Or, more complex decisions involving the evaluation of sets of rules might be more readily developed through prototyping in a knowledge-based environment such as Goldworks III that supports "frames". In all cases, because each of the three environments can be integrated with the others, it is possible to integrate components of the overall system that have been developed through prototyping using different software environments. With this capability in mind, it is possible for an underlying prototyping strategy to be implemented efficiently.

Some Categories of Special Consultive Needs for Research Practitioners. In identifying priorities associated with developing a consultive system, our particular interest is in focusing our efforts on the elements of the research process that experienced researchers emphasize (and which novices would be unaware) rather than those emphasized in textbooks. Although beyond the scope of this paper, some of these address broader aspects of research as inquiry such as (a) planning and implementing programmatic research (vs a single-study perspective), (b) strategies for networking with researchers of similar interests, (c) methodological approaches to qualitatively analyzing (e.g., observing, measuring, describing) both problem settings and treatment effects, (d) procedures for evaluating trends in research literatures and identifying (in specific areas) parallel research developments by non-educational researchers, and (e) standards for designing research studies given intended conclusions and for interpreting conclusions from studies given design features. Although our work is in the formative stages, we feel that it is possible to develop expert systems software that provides at least minimal guidance for inexperienced researchers in usable modular forms that could eventually be encompassed within a single program. At the same time, we also envision working to develop applications that support the more traditional research domains such as principles for test development, statistical analysis, and traditional research design.

Representation of Research Expertise: Examples of Proficiency Combining Textbook-Based and Non-Textbook Knowledge

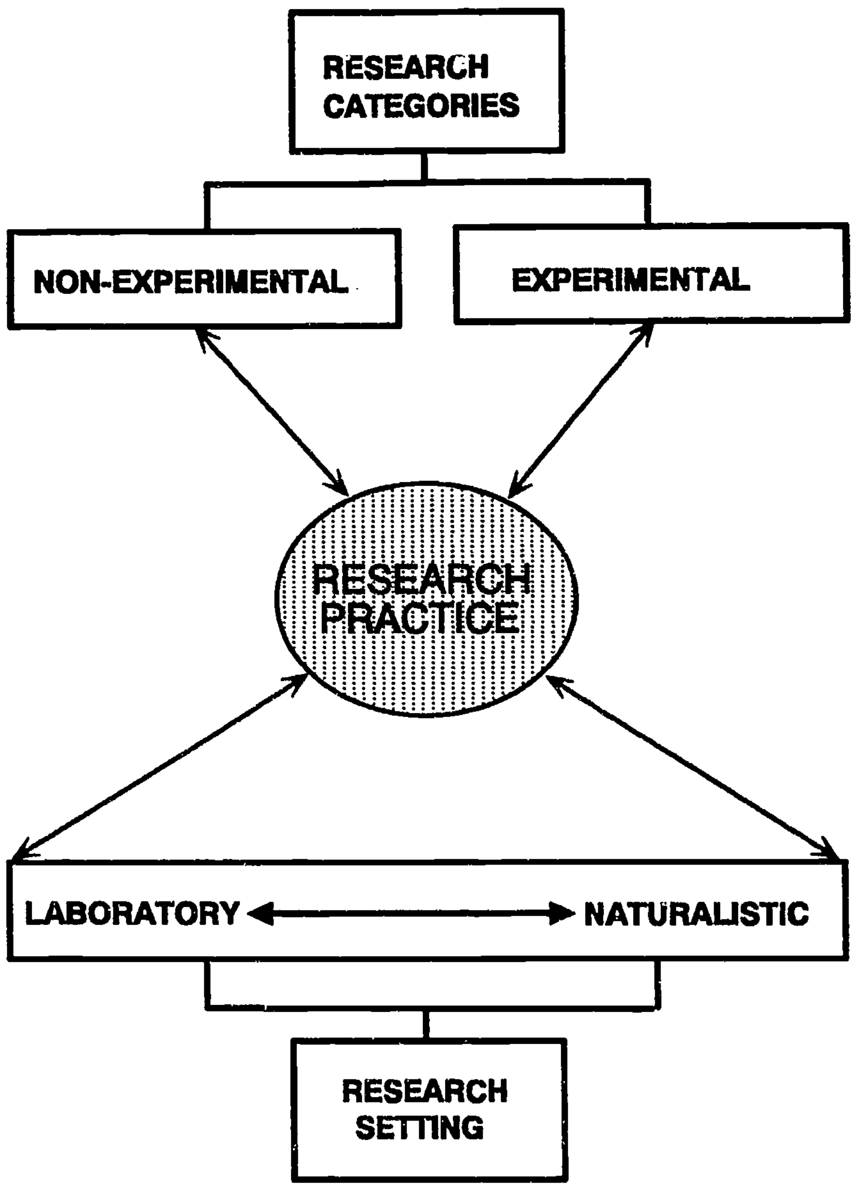
This section provides some examples (see Figures 5, 6, 7, 8, and 9) illustrating the representation of research expertise in forms that are useful in the development of expert systems. In doing so, the forms of expertise considered reflect recent perspectives on the development of knowledge-based cognitive skills for problem-solving from the fields of instructional design (e.g., Carnine & Kameenui, 1992; Foshay, 1991) and cognitive science (Anderson, 1982, 1984; Resnick, 1989).

Implications of Expert Systems Methodology for Educational Research

There are a number of important implications that follow from the application of expert systems technology to the problem of providing consultive support to research practitioners. These are:

* The best use of such a system (and the best environment for development) would be in conjunction with the type of extended apprenticeship model described by Romance & Vitale (1993) for developing action research skills of school personnel. Within this context, a major goal would be to capture different aspects of research expertise across the range of possible research applications as they are being supported. Although we can envision a useful collection of modular or "stand-alone" expert systems emphasizing different key aspects of the research process in a limited fashion, we recognize that the overall concept of capturing all research expertise within a single software application having great levels of detail is unrealistic for the immediate future. Thus, we believe it is presently feasible to develop limited expert systems that research practitioners in schools would find useful.

RESEARCH PROBLEM DOMAIN: DIMENSIONS FOR CLASSIFICATION



TYPES OF RESEARCH: A CONCLUSION-BASED DISTINCTION

INFORMATION AS PRODUCT OF RESEARCH

INFORMATION AS KNOWLEDGE IN THE FORM OF IF/THEN RULES THAT ALLOW PREDICTION OR CONTROL

KNOWLEDGE THAT ALLOWS PREDICTION COMES FROM IF/THEN RELATIONSHIPS OBTAINED FROM CORRELATIONAL STUDIES	KNOWLEDGE THAT ALLOWS CONTROL COMES FROM IF/THEN RELATIONSHIPS OBTAINED FROM EXPERIMENTAL STUDIES
INF. FORM: GIVEN X PREDICT Y	INF. FORM: DO X TO CAUSE Y

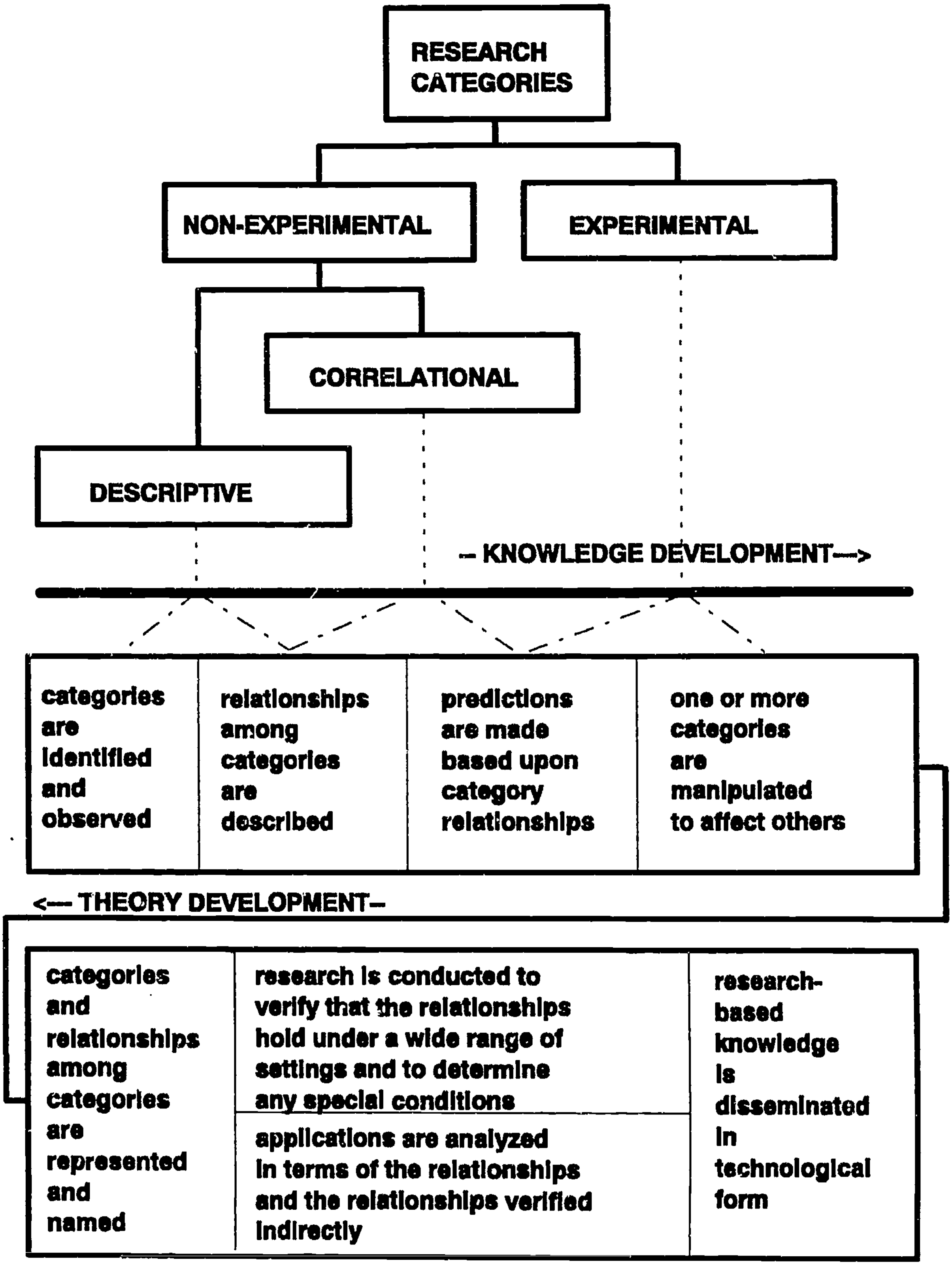
KNOWLEDGE VALIDITY TESTS

RELATIONSHIPS MUST BE DISCONFIRMABLE IN TERMS OF FUTURE OUTCOMES THAT OCCUR OR DO NOT OCCUR

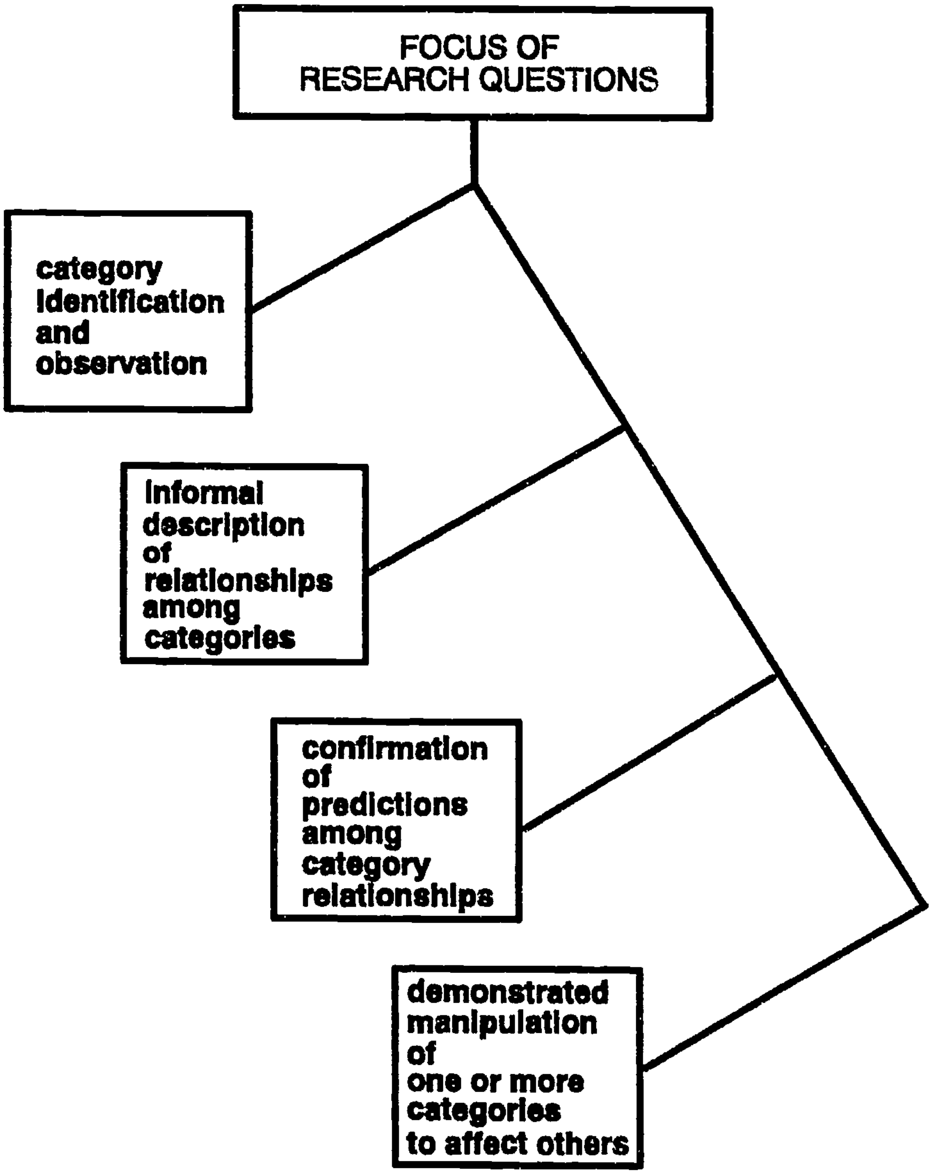
THE RESEARCH COMMUNITY MUST BE ABLE TO USE DEFINITIONS OF X AND Y TO EITHER MAKE PREDICTIONS OF Y FROM KNOWLEDGE OF X ... OR... TO PRODUCE Y FROM MANIPULATING X

VITALE
AERA 1993
Figure 7

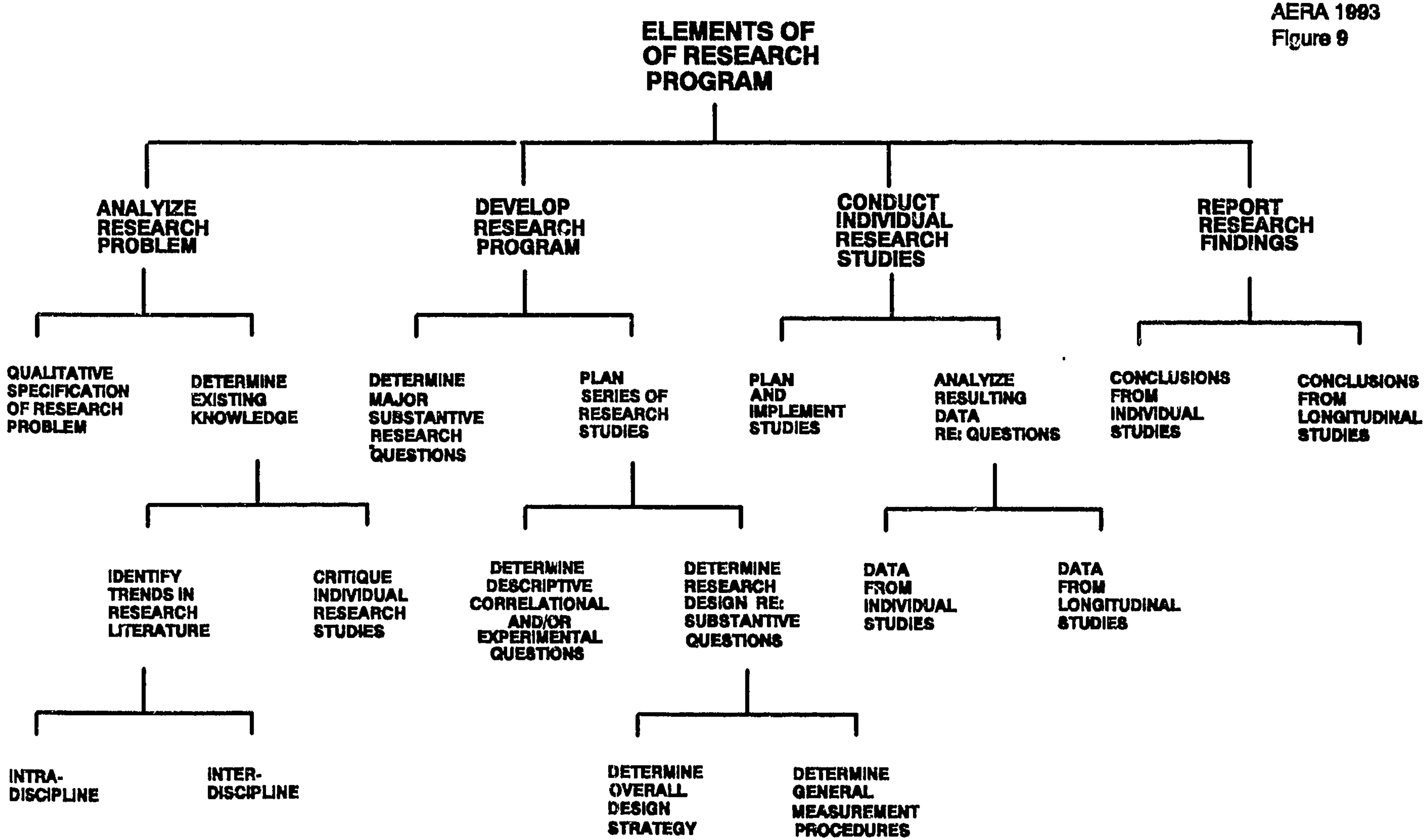
TYPES OF RESEARCH AND THE RESEARCH PROCESS



CATEGORIES OF EMPIRICAL RESEARCH



VITALE
AERA 1993
Figure 9



* It is clear to us that any serious attempt to represent expertise within the field of research would eventually lead to an improved reformulation of how methodological principles and practices in research are represented. At the present time, just the application of parsimony alone would help eliminate the excessive terminology (and "social jargonism") that professionals new to the field must face. However, even beyond verbal streamlining, any serious attempt to address the question of research methodology as procedural rather than declarative knowledge would have the effect of refocusing attention upon what the purpose of research is within the field and how it should be conducted rather than upon self-serving verbal discourse that serves as an end in its own right.

* Finally, it is also clear that having the capability to develop knowledge-based systems that provide expert consultation should also have an effect on how future researchers can be trained in general. Specifically, having the capability to represent research expertise as standards of practice across the domain of research problems would directly support the development of instructional models (e.g., Lawler, 1987; Polson & Richardson, 1988; Schank, 1991) from the field of artificial intelligence, including intelligent (i.e., knowledge-based) tutoring systems in which students would receive computer-based guidance as they apply methodological skills to solve problems or engage in simulations to gain higher-order research capabilities. In effect, the development of expert systems, even limited ones, would open the gateway for dramatic improvements in training novice educational researchers, both in and out of school settings.

References

Anderson, J. R. (1982). Acquisition of cognitive skill. Psychological Review, 89(4), 369-406.

Anderson, J. R. (1987). Skill acquisition: Compilation of weak-method problem solutions. Psychological Review, 94(2), 192-210.

Bogdan, R.C., & Biklin, S.K (1992). Qualitative research for education. Boston: Allyn & Bacon.

Carnine, D. (1992). Introduction. (pp 1-22). In D. Camine & E.J. Kameenui (Eds.), Higher order thinking. Austin, TX: Pro-Ed.

Carr, C. (1992). Performance support systems: A new horizon for expert systems. AI Expert, 44-49.

Foshay, W.R. (1991). What we know (and don't know) about training of cognitive strategies for technical problem-solving. (pp 344-353). In G.A. Anglin (Ed.), Instructional Technology. Englewood, CO: Libraries Unlimited.

Jackson, P.W. (1990). The functions of educational research. Educational Researcher, 3-9.

Johnson, J.M., & Pennypacker, H.S. (1980). Strategies and tactics of human behavioral research. Hillsdale, N.J.: Earlbaum.

Kerlinger, F.N. (1986). Foundations of behavioral research. N.Y.: Holt.

Kuhn, T.S. (1970). Structure of scientific revolution. Chicago, ILL: University of Chicago Press,

Lawler, R.W. (1988). Learning environments: Now, then, someday. (pp 1-26). In R.L. Lawler & M. Yazdani (Eds.), Artificial Intelligence and Education (Vol. 1). Norwood, N.J.: Ablex Publishing.

Luger, G.F., & Stubblefield, W.A. (1989). Artificial intelligence and the design of expert systems. Redwood City, CA: Benjamin-Cummings.

Marshall, C., & Rossman, G.B. (1989). Designing qualitative research. Newbury Park, CA: Sage Publications.

Payne, E.C., & McArthur, R.C. (1990). Developing expert systems: A knowledge engineer's handbook for rules and objects. N.Y.: Wiley.

Polson, M.C., & Richardson, J.J., (Eds.). (1988). Foundations of intelligent tutoring systems. Hillsdale, N.J.: Earlbaum.

Riesbeck, C., & Schank, R. (1989). Inside Case-Based Reasoning. Hillsdale, N.J.: Earlbaum.

Resnick, L. B. (1989). Introduction (pp 1-24). In L. B. Resnick (Ed.) Knowing, learning, and instruction. Hillsdale, N. J.: Erlbaum.

Romance, N.R., & Vitale, M.R. (1993, April). Evolution of a 5-year action research project: Results and implications for empowering school improvement. Paper presented at the Annual Meeting of AERA, Atlanta, GA

Schank, R.C. (1991). Case-based teaching: Four experiences in educational software design. Northwestern University: Institute for the Learning Sciences.

Shaw, J., & Zeichick, A. (1992). Expert system resource guide. AI Expert, 7(12), 42-49.

Sidman, M. (1960). Tactics of scientific research. N.Y.: Basic Books.

Tesch, R. Qualitative Research. Falmer Press, 1990.

Vitale, M.R. (1992, January). Expert systems as research tools. Paper presented to the Annual Conference of the North Carolina Center for Research in Education, Chapel Hill, N.C.

Vitale, M.R. (1993, January). Complementary relationships among qualitative and quantitative research: Methodological perspectives from behavior analysis. Paper presented to the Annual Conference of the North Carolina Center for Research in Education, Greensboro, N.C.

Winston, P.H. (1992). Artificial intelligence. (3rd ed.). Reading, MA: Addison-Wesley.